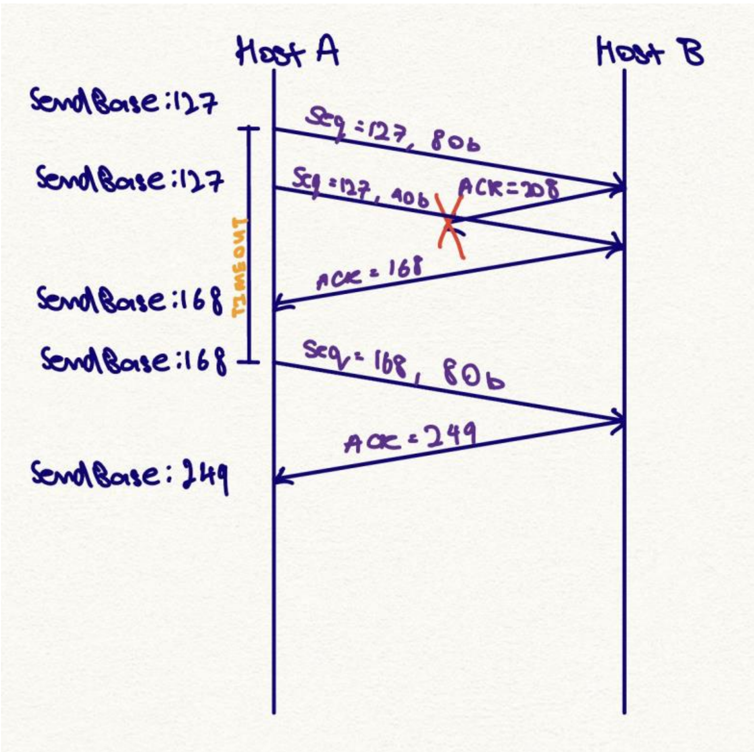
1. Hosts A and B are communicating over a TCP connection, and Host B has already received from A all bytes up through byte 126. Suppose Host A then sends two segments to Host B back-to-back. The first and second segments contain 80 and 40 bytes of data, respectively. In the first segment, the sequence number is 127, the source port number is 302, and the destination port number is 80. Host B sends an acknowledgement whenever it receives a segment from Host A.

A. In the second segment sent from Host A to Host B, what are the sequence number, source port number and destination port number.
    a. Sequence number: 208
    b. Src Port Number: 302
    c. Des Port Number: 80

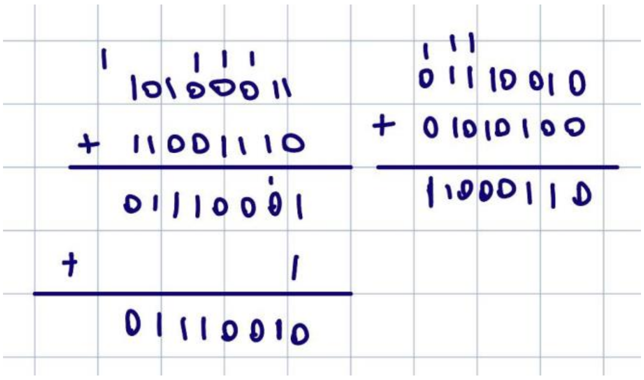
B. If the first segment arrives before the second segment, in the acknowledgment of the first arriving segment, what is the acknowledgment number, source port number and destination port number? What can you say about the sequence number of this acknowledgement segment?
    a. Src port number and des port number remain the same, else TCP connection is broken
    b. ACK = 208
    c. Sequence number stays the same since ACK contains no data and therefore does not need to “consume” a sequence number

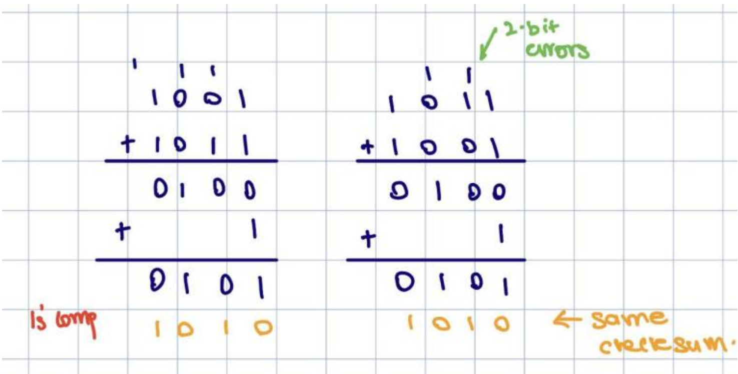
C. If the second segment arrives before the first segment, in the acknowledgment of the first arriving segment, what is the acknowledgment number?
    a. ACK = 168

D. Now, suppose that the two segments sent by Host A arrive in order to Host B. The first acknowledgment is lost and the second acknowledgment arrives after the first time-out interval. Draw a timing diagram, showing these segments and all other segments and acknowledgements sent. You can assume that there are no other packets lost. For each segment in your diagram, provide the sequence number and the number of bytes of data; for each acknowledgment, provide the acknowledgment number.
    a.



2. UDP and TCP use 1s complement for their checksums. Suppose you have the following three 8-bit bytes: 10100011, 11001110 and 01010100. What is the 1s complement of the sum of these 8-bit bytes? (Note that we use only 8 bits in the exercise, although the real checksum is 16 bits)

A. show all the phases in the calculations clearly
   a. first, sum the 3 numbers:
      i. 

      ii. There is an overflow, which is added back to the sum at the end as shown
      iii. The answer for the summing is: 11000110
   b. Take the one's complement of the answer:
      i. Simply swap any 1 to 0 and any 0 to 1 → answer is: 00111001
B. With the 1s complement scheme, how does the receiver detect errors?
   a. Sender side:
      i. Segment contents are treated as sequence of 16-bit integers
      ii. All segments are summed
      iii. Checksum: 1's complement of the sum
      iv. This checksum is sent in the TCP/UDP checksum field
   b. Receiver side
      i. Calculate checksum
      ii. All segments summed and added with the sender's checksum
      iii. Check that any 0 bit is presented in checksum
         1. If the checksum contains any 0, then error is detected and packet is discarded
C. Is it possible that a 1-bit error goes undetected? Show example or counterexample
   a. No, any 1-bit error will be detected.
   b. In the sum, the 1-bit error will make a 1 become a 0 and a 0 become a 1
D. How about a 2-bit error? Show example or counterexample.
   a. In some cases, 2-bit errors can go undetected
   b. If the last digit of the first bitstring is converted to a 0 and the last digit of the second bitstring is converted to a 1
   c. 

3. Host A and Host B are directly connected with a 80 Mbps link. There is one TCP connection between the two hosts, and Host A is sending to Host B an enormous file over this connection. Host A can send its application data into its TCP socket at a rate as high as 100 Mbps but Host B can read out of its TCP receive buffer at a maximum rate of 40 Mbps. Describe the effect of TCP flow control, i.e., how the receiver (Host B) consumes (i.e., uses) the data and how does the sender adjust its transmission?

- The main idea behind TCP flow control is that the receiver controls the sender, so that the sender will not overflow the receiver's buffer by transmitting too much too quickly.
- This is done by the receiver "advertising" free buffer space by including the receiver window value in the TCP header of receiver-to-sender segments
- The sender then limits the amount of unacknowledged data to receiver's receiver window value
- This guarantees the receive buffer of host B will not overflow.

4. Consider the attached figure about TCP transmission. x axis shows the transmission rounds (each lasts approximately one round trip time), y axis shows the congestion window size as number of segments.

Assuming TCP Reno is the protocol experiencing the behavior shown above, answer the following questions. Justify your answers with a short discussion.

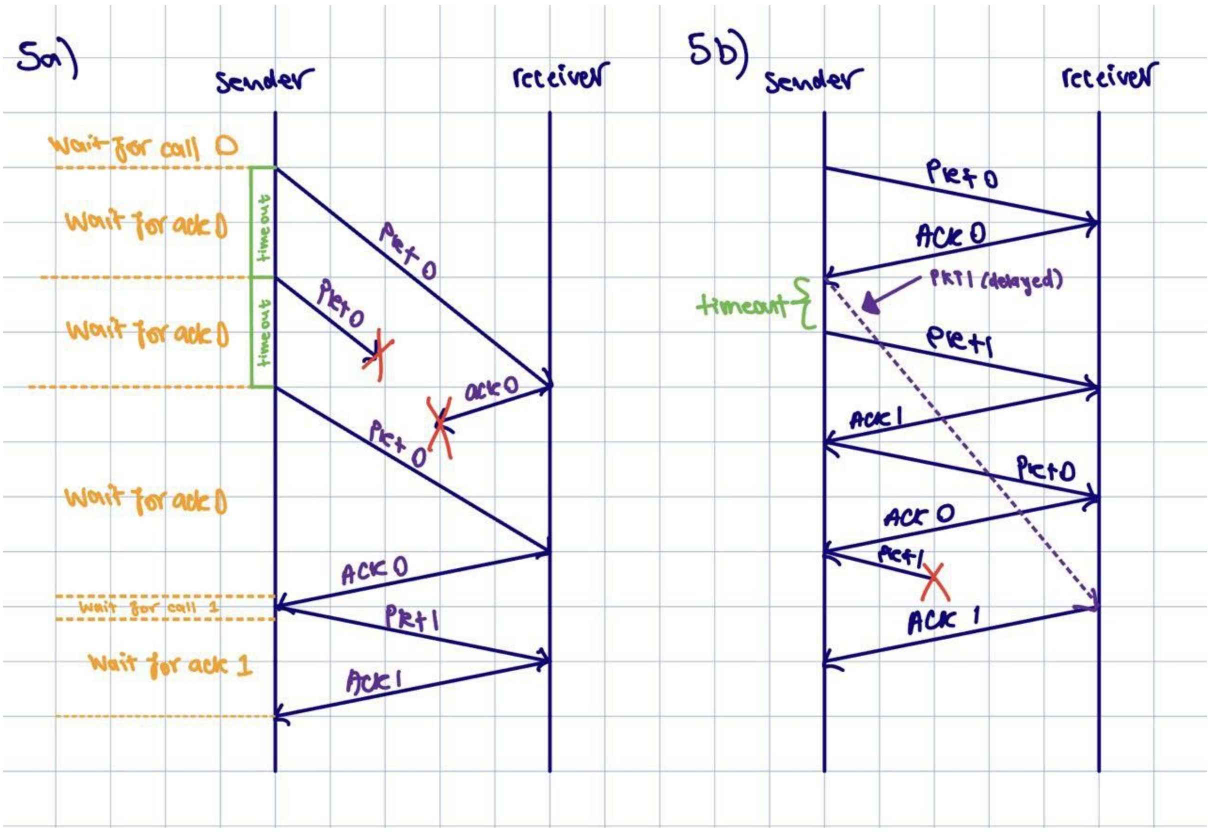
A. Identify the intervals of time when TCP slow start is operating
    a. TCP slow start operates between rounds 1 and 6, and between 23 and 26
B. Identify the intervals of time when TCP congestion avoidance is operating
    a. Between 6 and 16, and 17 and 22
C. After the 16th transmission round, is segment loss detected by a triple duplicate ACK or by timeout?
    a. Segment loss is detected by a triple duplicate ACK since the congestion window size started from threshold value and continued linearly
D. After the 22nd transmission round, is segment loss detected by a triple duplicate ACK or by timeout?
    a. Loss was detected by a timeout since the congestion window size started from 1 MSS
E. What is the initial value of ssthresh at the first transmission round?
    a. 32
F. What is the value of ssthresh at the 18th transmission round?
    a. 42/2 = 21
G. During what transmission round is the 70th segment sent?
    a. The 70^{th} segment is sent in transmission round 7
H. Assuming a packet loss is detected after the 26th round by the receipt of a triple dublicate ACK, what will be the values of the congestion window size and of ssthresh?
    a. 4(8/2) = 16

5. Consider the rdt 3.0 alternating bit protocol. Answer the following questions:

a. Draw a diagram (like in Lecture 5 "rdt3.0 in action"), where you show messaging between a sender and a receiver. Also, mark the states of the automata in the diagram (see lecture 5). In your diagram, you must at least express the situation, where a data packet sent is lost, and also the acknowledgment of the earlier packet, which is sent after the loss of the mentioned data packet, gets also lost

b. Show an example, where the rtd3 alternating bit protocol does not work correctly. This happens, if the channel between the sender and receiver can reorder messages, so that the messages sent from one end arrive to the other end in different order. The protocol has not been designed for this and thus works incorrectly occasional if this happens. Draw a diagram where you show the messages passed between sender and receiver. Draw the sender on the left and receiver on the right, while time axis runs down the paper. Now draw the data and acknowledgment messages as arrows (that take at least one time unit to pass) from sender to receiver. Indicate the sequence numbers with the messages clearly.

c. Answer for both:



6. Are the following statements related to TCP correct or incorrect? Justify your answer.

a. Host A sends a large file to host B over a TCP connection. Host B has nothing to send to A, not even acknowledgments since B cannot associate them to any data packet.
    a. False.
b. The size of the recipient window (receive window, rcwd) never changes during a TCP connection.
    a. False, the size of rcwd changes dynamically according to transmission
c. Assume that A sends a large file to B over a TCP connection. The number of bytes sent but not acknowledged by A cannot exceed the size of the recipient window (i.e., the receive buffer).
    a. True, the number of bytes sent should not exceed the rwnd, but in the case that it does, the cwnd of the sender is adjusted to manage flow control [TCP flow control]
d. Assume that A sends a large file to B over a TCP connection. If at a certain point the sequence number of the segment is m, then the sequence number of the next segment is m + 1.
    a. False, the sequence number of the following segment is m + size(current_segment)
e. Assume that the last value of SampleRTT in a TCP connection is 1 s. In this case, the current value of TimeoutInterval is certainly ≥ 1 s. Show the calculation.
    a. False
    b. TimeoutInterval = Estimated RTT + 4 * Dev RTT